Regression vs Neural Nets for Housing Prices
COMP 562 Final Project

Charles Tang
Jackson Witzke
Sarah Chipman
Austin Tinsley

Abstract
The goal was to be able to determine whether more complex models were needed in order to predict housing prices effectively. We used various linear regression techniques as an idea of a simple model, and a neural network as a more complex model. To compare these methodologies, we tested the more advanced models against a simple linear regression and noted the percentage decrease in loss. It was found that our neural network model had a larger decrease in percentage of loss against the linear regression. This leads us to the conclusion that more complex models were, in fact, worth the trouble in terms of accurately predicting house prices.

1. Introduction
There has been a massive increase in research and development towards machine learning and creating best techniques. As research in machine learning has progressed there has been drastic increase in complexity of techniques from simple linear regression to deep neural networks. The purpose of the development of these complex prediction models were to optimize efficiency and solve more convoluted problems. However, for simple problems are these complex models needed or is there a diminishing return that much simpler models could fill.

1.1. The Problem
Predicting housing prices is a problem that can be solved relatively easily using machine learning. At onset, we believed that the natural behavior of sale price of houses would be modeled well linearly. While our sale price data is not normally distributed, when we took the log of sale price we were able to produce more normal data, as shown in two graphs below. This allowed for linear models to work well with housing prices. Given this, we wondered if a neural net would even be necessary, especially since our initial results with linear regression did well.

University of Pennsylvania, CIS 419/519 Course Project.
Copyright 2017 by the author(s).
1.2. Literature Review

The idea of predicting housing prices is not an unusual or innovated application for machine learning. Information on the housing market can be as valuable as that of the stock market when applied correctly. Many researches in the past have tried to develop highly accurate models using regression techniques before the advancement of neural networks. One of the more successful regression techniques done in research to predict housing prices is that of creating a multiple regression model through model averaging. In research conducted by Koramaz they were able to take a multiple regression model that emphasized features based specifically to that of the location of a house in order to predict the price of a house. This model, then revealed that multiple regression models yielded low error, and that location-based features were highly correlated to that of the price, which is something that will be looked into in this paper (Koramaz & Dokmeci, 2012).

On the other hand, neural networks, specifically artificial neural networks (ANNs), are also commonly used for predicting housing prices. One paper touched on the importance of including entropy-based weighting to ensure features had reasonably weighted variables when modelling housing prices in Hong Kong. Results showed that the ANN enhanced with entropy based weighting served a more desirable function of the housing price with suitable input variables and relatively smaller sample sizes. (Chun Lin & Mohan, 2011) This lead us to believe that with appropriate application and usage of data, we could train a neural net to outperform simpler linear regression models. Further reading on comparisons between ANNs and other regression models preempted our findings in this paper. A study conducted on housing prices in Amherst, New York, compared mass appraisal of housing prices on three different models, an ANN, a multiple regression model (MRM), and an additive nonparametric regression model (ANR). It was found that ANN models are reliable and cost-effective methods for mass appraisal of residential housing. (Chun Lin & Mohan, 2011) The ANN model results in significantly smaller prediction errors than the MRM or the ANR model because of the many interactive terms between the independent variables underlining the ANN architecture. (Chun Lin & Mohan, 2011) While we are comparing different models than were introduced in this paper, it hints that the robustness of neural networks to map correlations between features may (and did) result in better predictions.

1.3. Hypothesis

In order to solve this problem it was decided that two different methodologies will be developed. The first using advanced linear regression techniques in order to determine which of the simpler machine learning techniques could derive the best results as represented by minimal mean square error. The second methodology would include that of a three layer neural network. The resulting errors will then be compared to simple linear regression in order to evaluate effectiveness. We predict the neural networks will achieve a lower error.

1.4. Preprocessing

Data pre-processing encompassed a lot of our preliminary work. Pre-processing considerations included what to do with outliers and how to handle classification data. We used different techniques for different methods, which will be explained in more detail below.

In methodology one, first we examine the Ames dataset for any obvious outliers. We remove two outliers based on the GrLivArea variable. We then plot the sale price and see that it is left skewed. Since linear regression works best on a normal distribution we take a log to normalize the data. Now we examine how much data is missing. We see that there is a lot of N/A data that should be filled in with None as a classifier or 0 numerically. We then encode all of our categorical variables with one hot encoding. Finally, we split our data set into training and testing data.

For methodology two, the same Ames dataset was taken and manipulated differently than that of methodology one, as a prospect of having more complex data manipulation to be integrated as having a more complex model in total. The non-numerical features were removed. Many of these non-numerical features were associated with that of numerical features, such as whether a house has a garage, and the square feet of the garage. Then outliers in the training data were removed by using an isolation forest to detect and remove the outliers. Lastly, the remaining thirty-eight numerical features were scaled using the feature-scaling method Min-Max Scaler in order to normalize the data, so it can be better evaluated by the neural network.

2. Baseline Results

After doing the data processing, we applied a simple linear regression model with no regularization to each of the sets of data. Our goal here was to get a baseline to compare against more advanced and tuned models. The simple linear regression model we used for methodology 1 reported a decent MSE of about 0.0154 whereas the simple linear regression model in methodology 2 reported a much better MSE of 0.0027. The difference in these MSEs is accounted for by the data processing done beforehand that will stay consistent for each model as we continued onto more complex models.
3. Method

3.1. Approach

For methodology one, we tried multiple different linear regressions and penalties. We began with a simple linear regression. Then, as we had initially decided, we tried the gradient boosting along with two other gradient boosting packages: LGB (Light Gradient Boosting), and XGB (eXtreme Gradient Boosting). We also tried Lasso, Elastic Net, and Kernel Ridge Regression. We can apply a Bayesian Average to the model parameters to account for the uncertainty in linear regression. (Raftery et al., 1997) We see that this Mean model achieves the lowest MSE on the test data. For methodology two it was decided that a neural network would be used as the example of a more complicated model in order to solve the problem of predicting housing prices. It was decided that a simple neural network with three layers, two densely connected layers and an output layer. In order to find the optimum activation function for the two hidden layers both ReLu and LeakyReLu activation functions were trained, tested, and compared. Both were trained originally using 500 epochs. It was noted that the model stopped improving in much less than 500 epochs, so an early stopping callback was applied. After this application the ReLu neural network was done training at a little over 200 epochs, and the Leaky ReLu neural network was done at around 160 epochs.

3.2. Results

The initial simple linear regression achieved an MSE of 0.0154. Next highest was normal gradient boosting which achieved an MSE of 0.0139, LGB got 0.0135, XGB and Kernel Ridge Regression tied at 0.0133. Elastic Net and Lasso both achieved an MSE of 0.0125. The averaged model achieved an MSE of 0.0119. This is a 4.8% improvement from the next best models we had (Elastic Net and Lasso), and a 23% improvement over our simple model. After both neural network models were trained, they were then evaluated against the testing data set aside on how well they predicted the housing price. This evaluation was also done using mean squared error. It was found that neural network using LeakyRelu outperformed the network using ReLu by having an MSE value of 0.002691; whereas, the ReLu model produced an MSE of 0.0008404.

The two methodologies were compared by how the best technique within each methodology did against the simple linear regression associated with the respective data manipulation done. This evaluation was done by looking at the percentage decrease in loss of the model compared to that of the simple linear regression. It was found that the advanced linear regression techniques only showed a percent decrease in loss of 22.7%, whereas, the neural network expressed a percent decrease in loss of 68.7%, which is roughly three times more of a percentage decrease than that of the advance linear regression methodology.

4. Conclusion

In conclusion it was determined that a more complex model does still provide significantly better results than that of a simpler model even when used to solve a simple problem when applying machine learning techniques, such as predicting housing prices. As expressed in the results the neural network using LeakyReLu activation functions outperformed the advance linear regression techniques by a factor of three.

It was a shock that LeakyReLu outperformed that of its more popular counterpart activation function that is ReLu because of the immense popularity. However, this is certainly not the first time that these results have appeared.
within a study. In previous research it was also found that LeakyReLu has outperformed that of ReLu across multiple datasets and network designs. This is more than likely due to the issue of gradients dying that is a common symptom of using ReLu, however, from a more theoretical aspect of the consistent successful of LeakyReLu over ReLu in research there needs more academic justification and research done. (Xu et al., 2015)

In contrast, we are unsurprised that the mean of our linear regression models out performed each individually. Each linear picks up on different features and regularizes them differently, and when we average them we get something of a combination of the best. Its not surprising that neural networks perform better than very complex linear regression functions. Neural networks can model non-linear relationships much better.

In the future, we would have liked to compare each of these models against a very basic linear regression model without any data pre-processing. Even the basic linear regression models we used as a baseline still had some data tweaks applied. Because the nature of the data played a large role in this problem, it would have been interesting to break down why certain data processing was desirable, and what the outcome of such changes were. The repository for this project can be found here https://github.com/comp526/Final-Project with the main models coming from the notebooks titled StackedRegression and NNet2.

**Acknowledgments**

Thanks to all who’ve helped us.

**References**


